UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/798,910	03/11/2004	Scott J. Broussard	AUS92031085US1	6996

45502 7590 12/11/2007

DILLON & YUDELL LLP
8911 N. CAPITAL OF TEXAS HWY.,
SUITE 2110
AUSTIN, TX 78759

EXAMINER	
HWA, SHYUE JIUNN	

ART UNIT	PAPER NUMBER
2163	

MAIL DATE	DELIVERY MODE
12/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/798,910	BROUSSARD, SCOTT J.	
	Examiner	Art Unit	
	James Hwa	2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 September 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 27 are pending in this office action. Claims 1, 10, 16 and 19 are amended, No claims added or cancelled. This action is responsive to Applicant's application filed on September 26, 2007.

Response to Arguments

2. Applicant's arguments filed 9/26/2007 have been full considered but are not persuasive.

Applicant argued that, Cantrill does not teach the added claimed limitation "detection of critical memory leak during execution of the software program" in claims 1, 10 and 19. The examiner respectfully disagrees.

Cantrill clearly teaches post-mortem memory leak detection process allows run-time considerations that are typically associated with detecting memory leaks (column 5, lines 3-5).

For the above reason, examiner believed that rejection of the last office action was proper.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the

international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 6, 7, 10, 15, 16, 19, 24 and 25 are rejected under 35 U.S.C. 102(e) as being anticipated by Cantrill (US Patent No. 6,523,141 B1, hereinafter "Cantrill").

As to claims 1, 10 and 19

Cantrill teaches

"A computer-implemented method for assisting the detection of critical memory leaks in a software program" as one standard method of identifying memory leaks involves looking at buckets of memory to identify a bucket which includes an abnormal number of allocations (column 2, lines 3-6).

"monitoring an amount of available memory for the software program during execution of the software program" as in one embodiment, iterating or walking through the root set may involve first walking through all module data, and then walking through all available thread stacks. Alternatively, walking through the root set may involve walking through all available thread stacks prior to walking through all available buffers (column 9, lines 42-48).

Cantrill teaches the amount of memory, both statically allocated memory and dynamically allocated memory, available to a computing system or, more specifically, processes executing on the computing system, is limited. As such, different processes must typically share memory resources. In order for memory to be shared, memory is recycled. In other words, a section of memory which is no longer used by one process may be reclaimed and allocated for use by another process (column 1, lines 14-21).

Cantrill also teaches the debugger may be configured to use the crash dump file to identify call-sites that are associated with memory leaks (column 5, lines 13-15). After beginning the execution of a debugger, the debugger may execute a command to locate memory leaks (column 5, lines 32-34).

Cantrill further teaches post-mortem memory leak detection process allows run-time considerations that are typically associated with detecting memory leaks (column 5, lines 3-5).

"determining if the amount of available memory for the software program is less than a predetermined amount" as walking through module data sections may involve walking through available buffers and available thread stacks such that some buffers are walked through before some thread stacks, and vice versa (column 9, lines 48-52).

Cantrill further teaches the available memory in a computing system is reduced; memory leaks generally cause an operating system to operate more slowly. When the overall amount of leaked memory is high, memory leaks may eventually result in no memory being available for allocations (column 1, lines 57-61).

"in response to such determination, storing a current stack walkback of each object currently referenced by the software program prior to the available memory dropping below an amount necessary to store the current stack walkback wherein the current stack walkback assists in the detection of a critical memory leak during execution of the software program" as a method for detecting a section of unreachable memory associated with a computing system includes obtaining a stack trace associated with the computing system. The stack trace is obtained from an image of an

operating system associated with the computing system. A call-site that has leaked memory is identified using the stack trace, and data structures, e.g., allocator data structures and module data structures, are walked through to identify a section of unreachable memory associated with the call-site (column 3, lines 26-34).

Cantrill also teaches once the stack traces are obtained, a determination is made as to whether the stack traces indicate that there is at least one existing memory leak. If the determination is that there does not appear to be any memory leakage, then the process of executing a command to locate memory leaks is completed (column 5, line 66 to column 6, line 4; see also figure 3).

Cantrill further teaches post-mortem memory leak detection process allows run-time considerations that are typically associated with detecting memory leaks (column 5, lines 3-5).

As to claims 6, 15 and 24

Cantrill teaches

"The step of performing the current stack walkbacks for the identified objects" as the stack trace is obtained from an image of an operating system associated with the computing system. A call-site that has leaked memory is identified using the stack trace, and data structures, e.g., allocator data structures and module data structures, are walked through to identify a section of unreachable memory associated with the call-site (column 3, lines 26-34).

Cantrill further teaches if it is determined that the obtained pointer is not associated with dynamically allocated memory, then process flow returns where a walk through the root set to identify pointers continues. That is, process flow returns where either a walk continues through the current module data section, if it contains additional pointers to be obtained, or a new module data section is walked through (column 7, lines 38-44).

As to claims 7, 16 and 25

Cantrill teaches

"The step of generating a statistics report including current stack walkbacks for the identified objects" as once the command to locate memory leaks is executed, a report on the memory leaks is obtained. The report on memory leaks may be subsequently used to facilitate the fixing of memory leaks. It should be appreciated that, in general, the contents and format of a report on memory leaks may be widely varied. However, the contents of a report on memory leaks typically identify the call-sites or functions, which leaked memory. After the report on memory leaks is obtained, e.g., by a user, the process of managing memory leaks is completed (column 5, lines 44-53; see also element 304 of figure 3 and element 420 of figure 4A).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2-5, 8, 11-14, 17, 20-23 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cantrill (US Patent No. 6,523,141 B1) as applied to claim 1 above, and further in view of Schumacher (US Patent Application No. 2005/0076184 A1, hereinafter "Schumacher") and Fu (US Patent No. 7,089,460 B2, hereinafter "Fu").

As to claims 2, 11 and 20

Cantrill teaches

"determining if any analysis property of the objects being referenced following a garbage collection process exceeds a predetermined limit for such analysis property, wherein the predetermined limit for an object's age is an object age limit" as a post-mortem memory leak detection process enables a system crash dump, or a file which essentially contains an image of the operating system of a system which has failed, to be used to locate memory leaks. By using data sections associated with the kernel as a root set, in addition to information contained in the crash dump file, a suitable garbage collection algorithm, e.g., a mark and sweep garbage collection process, may be used to identify memory leaks (column 4, lines 51-59).

Although Cantrill teaches the claimed limitations, Cantrill does not explicitly teach "monitoring a specified one or more analysis properties of the objects referenced by the software program, wherein the one or more specified analysis properties consists of at least one of an object's age and an object's instance count".

Fu teaches the memory usage data is enhanced by reducing the impact of old elements included in the memory usage data. Reducing the impact of old elements, in one exemplary aspect, comprises filtering old memory usage data elements as one technique for enhancing memory usage data. Averaging old memory usage data elements with neighbors are one technique for is one form of low pass filtering (column 2, lines 58-64).

Fu further teaches an exemplary cutoff weighting subroutine that simply discards old memory usage elements. Memory usage weighting subroutine precedes iteration through each usage data element begins. The first step in the loop is decision where a determination is made whether the current usage data element is older than a threshold time. In exemplary embodiment the threshold time is arbitrary and depends upon the nature of the leak to be detected (column 5, lines 59-67).

Cantrill does not explicitly teach, "Identifying any objects determined to have one or more analysis properties that exceeds that property's predetermined limit".

Fu teaches

the invention is directed to detecting memory leaks in computing environments; analyzing records of memory usage detect memory leaks by programs, processes, jobs, threads and other memory utilizing components (column 1, line 65 to column 2, line 2).

Fu further teaches a method of detecting a leak in a computer memory, comprising: obtaining memory usage data for a memory using component; computing the first derivative of the memory usage data; determining if the area under a curve

based on said first derivative of the memory usage data is greater than a predetermined value (claim 1).

Cantrill does not explicitly teach "the predetermined limit for an object's instance count is an object instance count growth value".

Schumacher teaches monitors the allocated memory levels at a predetermined frequency, and determines peak allocated memory levels. The invention is further configured to determine increases in the peak allocated memory levels. The present invention determines that a memory leak exists when it has counted a predetermined number of increases in the peak allocated memory level. The predetermined number of times is herein sometimes referred to as an Alarm Limit (page 2, paragraph 0018).

Schumacher further teaches a sampling rate frequency (page 4, paragraph 0052, 0053; see also elements 38, 40, 42 of figure 5).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill, Fu and Schumacher before him/her, to modify Cantrill's identifying any objects determined to have one or more analysis properties that exceeds that property's predetermined limit because that would allow a new and improved method, system and computer readable medium for detecting leaks in computer memory as taught by Fu (column 3, lines 5-7).

As to claims 3, 12 and 21

Cantrill does not explicitly teach "the step of calculating an object's age by timing a current period starting when the respective object was instantiated".

Fu teaches memory usage weighting subroutine begins and proceeds to looping where iteration through each usage data element begins, starting from the oldest element. The first in the loop is decision where a determination is made whether the current usage data element is older than a threshold time. If the current usage data element is older than the threshold time, processing where the impact of the usage data elements are reduced by averaging the magnitude of the current data element with its neighboring data elements (column 6, lines 20-30).

Fu further teaches an acceptable method of calculating the area under the curve is summing all the values of all of the data elements determined by the first derivative calculation (column 7, lines 6-8; see also elements 510, 520 of figure 5).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill and Fu before him/her, to modify Cantrill's identifying any objects determined to have one or more analysis properties that exceeds that property's predetermined limit because that would provide an even more accurate computation of minima points as taught by Fu (column 2, lines 20-21).

As to claims 4, 13 and 22

Cantrill does not explicitly teach "the step of calculating object instance count growth as the magnitude of growth of an object's instance count over a given time period".

Schumacher teaches the method calculates an estimated memory leakage rate. The method may perform this step by determining the difference between the current peak memory level and the initial peak memory level increase after the startup period or the second increase since time zero. Note, the first sampled memory level after the startup period will indicate an increase over the initial allocated memory when the program began (page 4, paragraph 0040).

Schumacher further teaches the method compares and determines whether the calculated memory leakage rate exceeds the minimum rate for alarm operating parameter (page 4, paragraph 0041).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill and Schumacher before him/her, to modify Cantrill's calculating object instance count growth because that would allow operative to detect memory leaks while the monitored application runs in a normal mode, not a special debug mode as taught by Schumacher (page 1, paragraph 0010).

As to claims 5, 14 and 23

Cantrill does not explicitly teach, "the step of monitoring comprises monitoring objects within a class designated for monitoring".

Schumacher teaches the monitoring agent determines allocated memory levels of a monitored application at a sampling frequency and determines when a then-existing peak memory level has increased. The monitored application is determined to be

leaking memory when a determined number of increases in the peak allocated memory level are detected (abstract).

Schumacher also teaches a Simple Network Monitoring Protocol (SNMP) interface, as well as other notification mechanisms. Reporting mechanism is configured to allow selection of zero or more notifications from the group, to be performed upon detection of memory leak in excess of the minimum rate for alarm (paragraph 0033).

Schumacher further teaches agent monitors a tracked or monitored application to determine allocated memory usage. Step may include the substeps of sampling the memory level at the frequency determined by SAMPLING_RATE operating parameter for each monitored application for which a memory leak has not already been detected paragraph 0037; see also element 34 of figure 5).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill and Schumacher before him/her, to modify Cantrill's monitoring objects within a class designated because that would allow selection of zero or more notifications from the group, to be performed upon detection of memory leak in excess of the minimum rate for alarm as taught by Schumacher (page 3, paragraph 0033).

As to claims 8, 17 and 26

Cantrill does not explicitly teach " the step of generating a web interface for user viewing of the statistics report at a computer display".

Schumacher teaches referring now to the drawings wherein like reference numerals are used to identify identical components in the various views, a computer having memory leak detection and reporting system. The system includes a monitoring agent for monitoring peak allocated memory levels at a predetermined sampling frequency (page 1, paragraph 0013; see also elements 16 and 18 of figure 1).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill and Schumacher before him/her, to modify Cantrill's generating a web interface for user viewing of the statistics report because that would allowing an alarm interface to record an alarm/response for later retrieval/review as taught by Schumacher (page 3, paragraph 0033).

5. Claims 9, 18 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cantrill (US Patent No. 6,523,141 B1) as applied to claim 1 above, and further in view of Cirne et al. (US Patent Application No. 2004/0078540 A1, hereinafter "Cirne").

As to claims 9, 18 and 27

Cantrill does not explicitly teach, "the software objects are Java objects".

Cirne teaches when Java first became popular, many programmers thought that they no longer had to worry about memory leaks because with Java the programmer simply creates objects and the JVM takes care of removing them when they are no longer needed (page 1, paragraph 0012).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Cantrill and Cirne before him/her,

to modify Cantrill's the software objects are Java objects because that would allow the garbage collector finds objects that are no longer needed by an application and removes them when they can no longer be accessed or referenced as taught by Cirne (page 1, paragraph 0012).

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James Hwa whose telephone number is 571-270-1285. The examiner can normally be reached on 8:00 – 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on 571-272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.
Status information for unpublished applications is available through Private PAIR only, for more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the PAIR system contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH
12/3/2007

James Hwa
Examiner
Art Unit 2163

DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100